Year: 2008

Using Nested Surfaces for Visual Detection of Structures in Databases

Mazeika, Arturas ; Böhlen, Michael Hanspeter ; Mylov, Peer

Abstract: We define, compute, and evaluate nested surfaces for the purpose of visual data mining. Nested surfaces enclose the data at various density levels, and make it possible to equalize the more and less pronounced structures in the data. This facilitates the detection of multiple structures, which is important for data mining where the less obvious relationships are often the most interesting ones. The experimental results illustrate that surfaces are fairly robust with respect to the number of observations, easy to perceive, and intuitive to interpret. We give a topology-based definition of nested surfaces and establish a relationship to the density of the data. Several algorithms are given that compute surface grids and surface contours, respectively.

DOI: https://doi.org/10.1007/978-3-540-71080-6_7

Originally published at:
Mazeika, Arturas; Böhlen, Michael Hanspeter; Mylov, Peer (2008). Using Nested Surfaces for Visual Detection of Structures in Databases. In: Simoff, Simeon J; Böhlen, Michael Hanspeter; Mazeika, Arturas. Visual Data Mining: Theory, Techniques and Tools for Visual Analytics. Berlin / Heidelberg: Springer, 91-102.
DOI: https://doi.org/10.1007/978-3-540-71080-6_7

Using Nested Surfaces to Detect Structures in Databases

Arturas Mazeika[1,2] Michael Böhlen[1] Peer Mylov[2]
arturas@cs.auc.dk boehlen@cs.auc.dk mylov@hum.auc.dk

[1] Department of Computer Science, Aalborg University, Fredrik Bajers Vej 7E, 9220 Aalborg, Denmark
[2] Institute of Communication, Aalborg University, Niels Jernes Vej 14, 9220 Aalborg, Denmark

Abstract. We define, compute, and evaluate *nested surfaces* for the purpose of visual data mining. Nested surfaces enclose the data at various density levels, and make it possible to equalize the more and less pronounced structures in the data. This facilitates the detection of multiple structures, which is important for data mining where the less obvious relationships are often the most interesting ones. The experimental results illustrate that surfaces are fairly robust with respect to the number of observations, easy to perceive, and intuitive to interpret. We give a topology-based definition of nested surfaces and establish a relationship to the density of the data. Several algorithms are given that compute surface grids and surface contours, respectively.

1 Introduction

Visual data mining exploits the human perceptual faculties to detect interesting relationships in the data. To support the detection of relationships it is important to visualize data in a form that is easy understandable to humans. It is common to use scatter plots for this purpose [3]. Employing scatter plots is intuitive as each observation is faithfully displayed. Scatter plots have successfully been used for detecting relationships in two dimensions. For higher dimensions scatter plots are combined with grand tour methods. A grand tour displays a smooth rotation of two dimensional projections that eventually covers the entire high dimensional search space.

Scatter plots hit limitations if the dataset is big, noisy, or if it contains multiple structures. With lots of data the amount of displayed objects makes it difficult to detect any structure at all. Noise easily blurs the picture and can make it impossible to find interesting relationships. Finally, with multiple structures it often happens that one structure is more pronounced than another. In this situation the less pronounced structures easily get lost. For the purpose of data mining this is particularly bad as it is usually the less obvious relationships that are the most interesting ones. Surfaces equalize the more and less pronounced structures and thus support the detection of less obvious relationships.

In this paper we explore the potential of nested surfaces to analyze data sets. Nested surfaces enclose the data at varying densities. Humans are used to perceive surface information and to abstract surfaces from individual observations. This greatly simplifies the interpretation of the data. Nested surfaces do not suffer if the amount of data is big, and the nesting supports the detection of multiple structures. We provide a topology-based definition of surfaces and prove that the boundary $\partial C_\alpha = \partial\{(x, y, z) : f(x, y, z) \geq \alpha\}$ is a surface if the density function f has a continuous derivative. This provides the basis for an algorithm that computes nested level surfaces. Given a density estimation, which has continuos derivative, and a density level α we give algorithms that compute surface grids and surface contours, respectively. The described methods have been implemented and integrated into the 3D Visual Data Mining (3DVDM) System. The 3DVDM System is used for explorative data analyses in a 6-sided Cave, a 180° Panorama, and on regular computer monitors. It can be downloaded from `http://www.cs.auc.dk/3DVDM` and runs on SGI and PC/Linux computers.

The nested surface method works well with continuous datasets that contain multiple structures. We expect that the method will also work fine for categorical data. In this case, the ordering of dimensions and other parameters may be significant and can yield different visual results.

Usually, high number of observations overloads scatter plots. In contrast, nested surfaces produce nice results. The visualization is not affected by a high number of observations and it is continuously improving as the number of observations increases.

The computation of nested surfaces for the purpose of data mining has only received scant attention. Mostly surfaces have been investigated in connection with advanced visualization techniques, such as rendering, lighting, transparency, or stereoscopy [5,8,9,13]. These approaches focus on methods and data structures related to visualization aspects. Our goal is the computation of the defining structure of surface that emphasize the structure of the data.

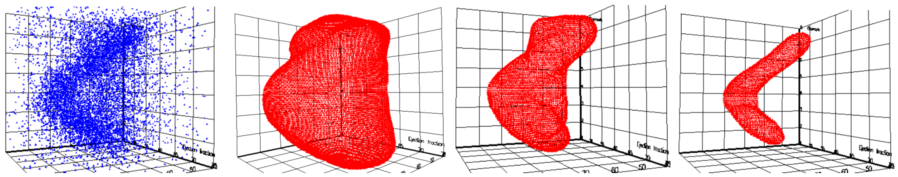
The paper proceeds as follows. We motivate our method in Section 2. Section 3 provides background material about probability density functions (PDFs), kernel estimations, clustering, and outliers. Section 4 defines surfaces. Section 5 gives algorithms for computing PDF estimates, level grid surfaces, and level contour surfaces. The algorithms are evaluated in Section 6. Section 7 discusses experimental results. Section 8 summarizes the paper and points to future work.

2 Motivation

Scatter plots are used to find structures in data. These structures are usually described as an accumulation of points. Scatter plots are good in getting a first impression of the data set, but they have a number of disadvantages. On one hand it is hard to understand very dense regions since data points hide each other. On the other hand it is also difficult to investigate sparse regions since data points in sparse areas are easily perceived as noise.

To illustrate our method we use the Spiral–Line data set presented in Figure 1(a). The data set consist of a vertical line in the middle (4'000 points), a spiral curve around the line (4'000 points) and uniformly distributed noise (2'000 points). The data points around the spiral curve form the most dense and notable region. Since the data points around the vertical line have a higher spreadness it is easy to treat it is as noise and not pay attention to it.

Figures 1(b) to 1(d) present the surfaces for different density levels α. Figure 1(b) shows the surface for the lowest density level. This Figure can be used for the detection of outliers. Figures 1(c) and 1(d) show surfaces for higher density levels. Together with Figure 1(b) they emphasize the structure of the data set. Note that the surfaces in Fig-



(a) Dataset (b) Surface for $\alpha = 1/10m$ (c) Surface for $\alpha = 5/10m$ (d) Surface for $\alpha = 9/10m$

Fig. 1: Spiral–Line DB and Associated Surfaces. m denotes the maximum density in DB

ure 1(c) clearly identify the vertical line *and* the spiral (the quality is much better on the monitor, see also Figures 5 and 6).

In contrast to scatter plot visualizations, surfaces do not deteriorate if the amount of observations increases. Nested surfaces are often easier to interpret than the raw data. Moreover multiple nested surfaces at different density levels facilitate the analysis of the data at different levels of detail. This gives the ability to explore the internal structure of data regions.

3 Preliminaries

3.1 Probability Density Function

Throughout, we assume that the data has been normalized to the three-dimensional unit cube, i.e., each coordinate falls into the [0,1] interval.

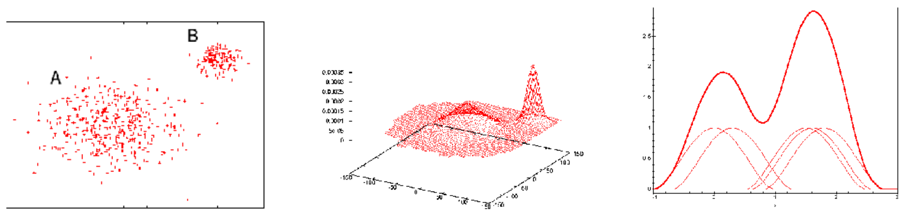
Definition 1. *(Probability density function) Let X be a 3 dimensional random vector with distribution function F. A 3-dimensional real value function f with*

$$F(x, y, z) = \int_{\infty}^{x} \int_{\infty}^{y} \int_{\infty}^{z} f(t, s, q)\; dt\, ds\, dq$$

is a probability density function (PDF).

Figure 2(a) shows a dataset with two clusters: A and B. The corresponding PDF is shown in Figure 2(b). The PDF shows the density of the dataset. Since the density of region A is lower than the density of region B the PDF value for region B is higher than for region A. The PDF also shows that region A is more spread than region B.

In general, we have to estimate the PDF because we are dealing with random datasets for which the true PDF is unknown. Different PDF estimates were proposed in the literature, with the kernel method being one of the most general ones [4,2,12,11,6]. The essence of the kernel method is that each observation increases the chances of having another observation nearby. Therefore, we draw a symmetric kernel with an area equal to 1 around each observation. Adding all kernels (cf. Figure 2(c)) yields an estimate for the PDF. To control the influence of one observation on the overall estimation



(a) Dataset D (b) PDF for Dataset D (c) Kernel Addition

Fig. 2: Dataset and Corresponding PDF

a smoothing parameter h is introduced. The kernel estimate [10] for a a set of observations, $(X_i, Y_i, Z_i), i = 1, \ldots, n$, at point (x, y, z) is defined as follows:

$$\hat{f}_K(x, y, z) = \frac{1}{nh^3} \sum_{k=1}^{n} K\Big(\frac{x - X_i}{h}, \frac{y - Y_i}{h}, \frac{z - Z_i}{h}\Big), \tag{1}$$

where K is a function (kernel) with $K \geq 0$, $\int K = 1$, and $K(x) = K(-x)$.

Various kernels K have been proposed in the statistical literature. Examples include square wave or Gaussian functions. It has been shown [12] that the accuracy of the estimation depends mostly on the smoothing parameter h and less on the choice of the kernel K. Parzen [2] showed that the smoothing parameter

$$h = h_{opt} = c(K, \sigma_1, \sigma_2, \sigma_3)/n^{-1/7} \tag{2}$$

minimizes the mean integrated square error (MISE):

$$\text{MISE} = \mathbf{E} \iiint (\hat{f}(x, y, z) - f(x, y, z))^2 \, dx \, dy \, dz. \tag{3}$$

c is constant for a given dataset and depends on the variance $(\sigma_1^2, \sigma_2^2, \sigma_3^2)$ of the random vector (X_1, X_2, X_3) and the kernel function K.

3.2 Clusters and Outliers

Density functions are also used to define clusters and outliers. Clusters and outliers are important characteristics of a dataset, and they are often used for data analysis. In the next section we will establish a relationship between clusters and surfaces. Let D be a set of 3 dimensional points, and let $(\mathbf{x}, \mathbf{x}^*] = \{\mathbf{x}t + \mathbf{x}^*(1-t), t \in (0, 1]\}$ be an interval in the three-dimensional space.

Definition 2. *(Cluster) A cluster for a set of local maxima M of the density function f and threshold ξ is the subset*

$$C = \{\mathbf{x} \in D \mid \forall \mathbf{x}^* \in M \wedge \forall \mathbf{y} \in (\mathbf{x}, \mathbf{x}^*] : f(\mathbf{y}) \geq \xi\}.$$

Definition 3. *(Outliers) The points $O \subseteq D$ are* outliers *iff for all local maxima $\mathbf{x}^*$ of the density function f*

$$O = \{\mathbf{x} \in D \mid \forall \mathbf{y} \in (\mathbf{x}, \mathbf{x}^*] : f(\mathbf{y}) < \xi\}.$$

Thus, a cluster is a set that contains PDF center (maxima) points together with all surrounding points that "exceed noise level ξ".

4 Surface Definition

We use a topological approach to define a surface. Intuitively, a surface is a set of points iff the neighbourhood of any point is similar to a two–dimensional open disk. To define the resemblance with an open disk we use a homeomorphic (one-to-one, continuous inverse) function.

Definition 4. *(Elementary surface) Let f be a function that maps an open disc D^2 to a set of points S. S is an* elementary surface *iff f is homeomorphic.*

Definition 5. *(Surface) A* surface *is a connected set of points iff the neighbourhood of any point of the surface is an elementary surface.*

Next, we establish a relationship between the border of a cluster and a surface. A border is a set of points: $\partial C = [C] \backslash C^\circ$ where $[C]$ contains the limit points of C and C° contains the inner points of C. We show that ∂C is a surface by giving a parametrisation function that maps a disk D^2 into ∂C.

Theorem 1. *(Implicit function theorem) Suppose $f : R^n \times R^m \to R^m$ is differentiable in an open set around (u, v) and $f(u, v) = 0$. Let M be the $m \times m$ matrix given by*

$$M = \left(\frac{\partial f_i(u)}{\partial x_{n+j}}\right) \quad 1 \leq i, j \leq m.$$

If det $M \neq 0$ then there is an open set $U \subset R^n$ that contains u and an open set V that contains v, such that for each $r \in U$ there exists $s \in V$ and $f(r, s) = 0$. If we define $g : U \to V$ as $g(r) = s$, then g is differentiable.

The implicit function theorem is a classical result and ensures the existence of a cluster boundary parametrisation. A proof can be found for example in [7].

Lemma 1. *Let f be a probability density function which has continuous derivative ($f \in C^1$), and C be a cluster for a set of maxima M and level noise ξ. Let grad $f(x) \neq 0, x \in \partial C$. Then ∂C is a surface.*

Proof. Notice that $\partial C = \{x \in D : f(x) = \xi\}$. Let $(a, b, c) \in \partial C$. Since grad$f \neq 0$ there is at least one coordinate x_i such that $\partial f / \partial x_i \neq 0$ at point (a, b, c). Then the implicit function theorem with $m = 1$ and $v = x_i$ proofs the lemma.

5 Algorithms

This section gives algorithms to compute nested surfaces: `Surface_GridPoints` and `Surface_GridLines`. Starting from the raw data, the first step is the estimation of the PDF. We scan the (sample of the) data set twice to estimate the PDF. The first scan is used to calculate the estimation parameters (cf. Equation (2)). The second scan is used to compute the actual PDF estimation. We use the Epanechninikov kernel [2], which is equal to 0 outside the area $t_1^2 + t_2^2 + t_3^2 \leq 1$. Thus, only observations that fall into the area $\{(t_1, t_2, t_3) : (t_1 - x)^2 + (t_2 - y)^2 + (t_3 - z)^2 \leq h^2\}$ influence the estimated PDF value at point (x, y, z).

Algorithm: PDF_Estimation

```
Input:
  Database with n observations: (X[i], Y[i], Z[i]),  i = 1,...,n
  Number of grid points in each dimension: g
Output:
  Data cube with PDF values on grid points: PDF
Body:
  1. Initialize PDF
  2. Calculate estimation parameters according to Equation (2)
  3. FOR i = 1 TO n DO
     3.1. Determine the set of PDF points A_g that are
          influenced by the data point (X[i], Y[i], Z[i])
     3.2. FOR EACH (k, l, m) ∈ A_g DO
               PDF[k, l, m] = PDF[k, l, m] + K((k-X_i)/h, (l-Y_i)/h, (m-Z_i)/h)
```

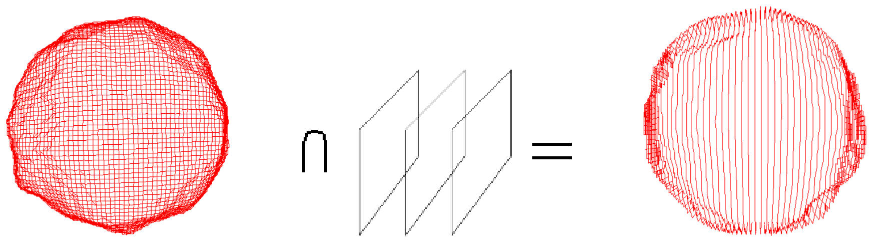
The `Surface_GridPoints` algorithm calculates the border $B = \partial\{(x, y, z) : f(x, y, z) \geq \alpha\}$. The basic idea of the algorithm is to scan the PDF and compare each value against its neighbours: if the value is greater than α and there exists a point in the neighborhood that is less than α then the value is added to B.

Algorithm: Surface_GridPoints

```
Input:
  Number of grid lines per dimension: g
  Data cube with PDF grid point values: PDF
  Density level: α
Output:
  Surface grid: B
Body:
  1. FUNCTION IsBorderPoint(PDF,i,j,k)
  2.    RETURN (PDF[i,j,k] ≥ α) AND (PDF[i',j',k'] < α)
                  for some (i',j',k') ∈ (i + h_1, j + h_2, k + h_3)
                  where h_1, h_2, h_3 = -1, 0, 1,  |h_1| + |h_2| + |h_3| = 1
  3. END FUNCTION
```

```
4. B = ∅
5. FOR i,j,k = 1 TO g DO
   5.1 IF IsBorderPoint(PDF,i,j,k) THEN B = B ∪ PDF[i,j,k]
```

The `Surface_GridLines` algorithm extends the `Surface_GridPoints` algorithm. The main idea of the algorithm is to draw contour curves on the surface. These curves, in turn, are calculated by intersecting a surface with cutting planes parallel to the XY, ZY, and ZX planes (cf. Figure 3).



(a) Surface

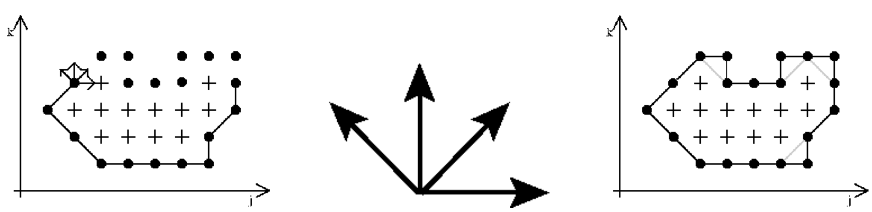
(b) Intersecting planes

(c) Vertical Grid-Lines-Surface

Fig. 3: Grid-Line-Surface

The idea of plane curve's calculation is presented in Figure 3. We scan the PDF values with a condition $i = i_0$ for ZY planes, $j = j_0$ for ZX planes, and $k = k_0$ for XY planes.

Figure 4(a) shows a cutting plane. Border points are shown as filled circles, inner cluster points as plus signs, and outer cluster points are not shown in the picture. The algorithm connects the border points to form a polygon curve.



(a) Connecting border points

(b) Drawing directions

(c) `Plane_Curve` method's output. Not valid directions are drawn in grey

Fig. 4: Curve computation in intercepting plane

For each PDF border point we are looking for PDF border points in the directions presented in Figure 4(b). Note, that we scan PDF from left to right, from bottom to top. Therefore, there is no need to draw lines to the bottom and to the left.

We make vertical and horizontal connections between border points. For diagonal we make additional checks. We do not draw diagonal line if there are two lines in its neighborhood (cf. Figure 4(c)). With this we avoid squares with crossing diagonals inside. The Individual steps of the ZY plain curve calculation are presented in the `ZY_Plane_Curve` algorithm.

Algorithm: ZY_Plane_Curve

Input:
  ZY plane number: i_0
  Data cube with PDF grid point values: PDF

Output:
  polygonal contour line on ZY plane at level i_0: $C = C_{i_0}^{ZY}$

Body:
  1. $C = \emptyset$, $i = i_0$
  2. FOR $j, k = 1$ TO g DO
     2.1 IF IsBorderPoint(PDF,i,j,k) THEN
       IF IsBorderPoint(PDF,i,j+1,k) THEN
         $C = C \cup \texttt{line}(i, j, k,\ \ i, j+1, k)$
       IF IsBorderPoint(PDF,i,j,k+1) THEN
         $C = C \cup \texttt{line}(i, j, k,\ \ i, j, k+1)$
       IF IsBorderPoint(PDF,i,j-1,k+1) AND
          ¬IsBorderPoint(PDF,i,j-1,k) AND
          ¬IsBorderPoint(PDF,i,j,k+1) THEN
            $C = C \cup \texttt{line}(i, j, k,\ \ i, j-1, k+1)$
       IF IsBorderPoint(PDF,i,j+1,k+1) AND
          ¬IsBorderPoint(PDF,i,j+1,k) AND
          ¬IsBorderPoint(PDF,i,j,k+1) THEN
            $C = C \cup \texttt{line}(i, j, k,\ \ i, j-1, k+1)$

Algorithm: Surface_GridLines

Input:
  Data cube with PDF grid point values: PDF

Output:
  Contour lines on the surface: C

Body:
  1. $C = \emptyset$
  2. FOR $i = 1$ TO g DO $C = C \cup \texttt{ZY\_PlaneCurve}(PDF, i)$
  3. FOR $j = 1$ TO g DO $C = C \cup \texttt{ZX\_PlaneCurve}(PDF, j)$
  4. FOR $k = 1$ TO g DO $C = C \cup \texttt{XY\_PlaneCurve}(PDF, k)$

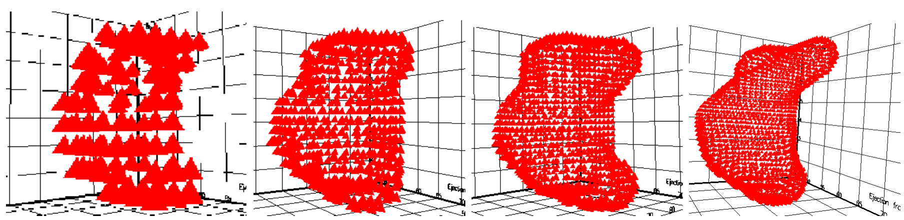
Note, that in order to include a 3D picture into the paper we have to project it into 2D. We use the `Surface_GridPoints` method to illustrate surfaces on a 2D devices while we use the `Surface_GridLines` method to illustrate surfaces in immersive 3D environments.

6 Evaluation

This section evaluates the quality of the algorithms numerically and visually. The experiments were calculated on the Pentium III 1GHz PC computer with 512MB of main memory running GNU/Linux OS with the gcc compiler.

6.1 Quality of the Surfaces

First, we evaluate the surface quality with respect to the number of grid lines. We use the three-dimensional scatter plot in Figure 1(a) and a single level surface for $\alpha = 1/10m$. In order to get a fair visual comparison of the influence of g on the quality of the surface we let the size of tetrahedra depend on g. It is chosen so that the tetrahedras visually are always near each other. Figures 5(a) and 5(b) show that $g = 10$ and $g = 20$ are not enough for a nice surface. There are too few tetrahedras at the ends of the spiral curve. As g reaches 30 the picture becomes detailed enough.



(a) $g = 10$ (b) $g = 20$ (c) $g = 30$ (d) $g = 50$

Fig. 5: Cluster Surface for $\alpha = 1/10m$ for Varying Values of g

To quantitatively measure the quality of the surfaces we use Equation (4). It quantifies the average error we make at any point (i, j).

$$AE_S = \frac{1}{g^2 \max_{x,y,z} \hat{f}_g(x, y, z)} \sum_{i,j=1}^{g} |\hat{s}_g(i, j) - s(i, j)|, \tag{4}$$

s is the parametrisation function that maps the open unit disk to $\partial C_\alpha = \partial\{(x, y, z) : f(x, y, z) \geq \alpha\}$. Since s is usually unknown we replace it with $\hat{s}_{\bar{g}}$ with a large value for $\bar{g}$:

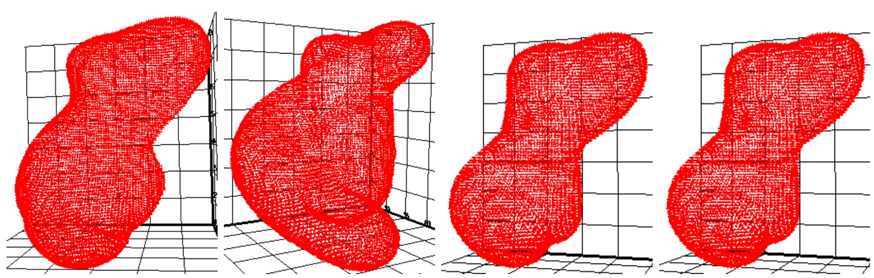
$$EAE_S = \frac{1}{g^2 \max_{x,y,z} \hat{f}_g(x, y, z)} \sum_{i,j=1}^{g} |\hat{s}_g(i, j) - s_{\bar{g}}(i, j)|, \tag{5}$$

Table 1 gives the numbers for EAE_S with $\bar{g} = 100$. The result shows that the error is very low. It is below 1% if the number of grid lines is greater than 30.

α	$g = 10$	$g = 30$	$g = 50$
1/10m	0.0289	0.0083	0.0045
5/10m	0.0249	0.0071	0.0038
9/10m	0.0069	0.0011	0.0005

Table 1: The EAE_S Error for Different Number of Grid Lines

Figure 6 presents the the impact of the size of the database sample on the surface quality. The figures show that $n = 10'000$ is sufficient for a nice surface. Note that Figure 6(b) is shown from a different prospective. This perspective emphasizes the unevenness of the vertical line.



(a) $n = 1'000$ (b) $n = 5'000$ (c) $n = 10'000$ (d) $n = 100'000$

Fig. 6: Cluster Surface for $\alpha = 1/10m$ and Varying Values of n

6.2 Space and Time Complexities

With the number of dimensions fixed at three the number of grid lines g and the number of observations n has the biggest impact on the computation time. We use the dataset from Figure 1(a) to measure the time to compute the surfaces.

Table 2 shows the times in seconds to calculate the surfaces from the raw data. A detailed analysis of the runtime reveals that the vast amount of the time is spent for the PDF estimation. Less than 1 second is needed to calculate a surface. Thus, to improve the performance it is possible to pre-compute and store PDFs. Table 3 shows that the size of the PDF is small and not usually relevant when compared to the size of the original database.

n	$g = 10$	$g = 30$	$g = 50$
1'000	<1	2	9
5'000	<1	6	24
10'000	<1	8	34
100'000	3	37	130
1'000'000	24	164	547

Table 2: Computation Time for Different Number of Grid Lines and Data Points

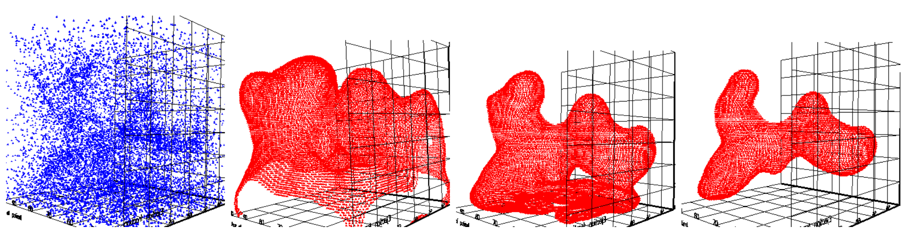
g	10	30	50	100
Size	4	108	500	4'000

Table 3: Size of PDF in KB

7 Experiments

This section illustrates our methods on an artificial data set (cf. Figure 7(a)). We show nested surface grids and offer an interpretation. Note that the visual information in the printed images is somewhat limited as three dimensions have to be projected into two. Also nested surfaces have to be shown in figures side-by-side. The reader may download and install the 3DVDM system to experiment with the surfaces.

The data set contains three data structures: 1) points, which are spread around randomly generated polygonal line, 2) a structure defined in terms of a simulated random variable: (uniform(0,1), uniform(0,1), exp(1)), and 3) uniform noise in the data set.



(a) Dataset (b) $\alpha = 1/10m$ (c) $\alpha = 3/10m$ (d) $\alpha = 5/10m$

Fig. 7: Artificial Dataset

It is hard to understand the structure from the scatter plot (cf. Figure 7(a)). The nested surfaces in Figures 7(b) to 7(d) emphasize and clarify the structure.

8 Summary and Future Work

In this paper we defined and evaluated nested surfaces for the purpose of visual data mining. Since humans perceive surfaces much easier than individual observations, our approach to data mining gives the ability to investigate the structure easily. In addition, surfaces clarify very dense and sparse (and the combination of both) regions of the data set. That gives an ability to detect arbitrary shaped structures in a data set.

The surface calculation is based on an estimated PDF which makes our method independent of the data. The PDF estimation is implemented as a three-dimensional cube. We presented empirical results that show that the space and time complexity is reasonable. It is possible to compute surfaces on the fly during data explorations. Real time interaction can be achieved by precomputing and storing small density estimates.

In the future we will refine our methods to find curves and 2-D structures in a data set. It would also be interesting to experiment with the display of visually advanced surfaces that use transparency, light and shading. Finally, we also want to develop a new data structure that enables interactive computation of the density surfaces.

Acknowledgments

This work is supported in part by the Danish research council through grant 5051-01-004. We greatly appreciate the comments of the 3DVDM project members and our partners from Nykredit. We thank the VRCN for the opportunity to work with immersive visualizations.

References

1. A. Mažeika, M. Böhlen, P. Mylov. Density Surfaces for Immersive Explorative Data Analyses. To appear in SIGKDD Workshop on Visual Data Mining, 2001.
2. B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, 1986.
3. D. A. Keim and H.-P. Kriegel. Visualization Techniques for Mining Large Databases: A Comparison. *Transactions on Knowledge and Data Engineering, Special Issue on Data Mining*, 8(6):923–938, 1996.
4. D. W. Scot. *Multivariate Density Estimation*. Wiley & Sons, New York, 1992.
5. E. J. Wegman, Q. Luo. Visualizing Densities. Technical Report Report No. 100, Center for Computational Statistics, 1994.
6. G. C. van den Eijkel, J. C. A. Van der Lubbe, E. Backer. A Modulated Parzen-Windows Approach for Probability Density Estimation. In *IDA*, 1997.
7. G. E. Bredon. *Topology and Geometry*. Springer–Verlag, 1995.
8. H. Shen and C. Johnson. Sweeping Simplicies: A Fast Isosurface Extraction Algorithm for Unstructured Grids, 1995.
9. J. Wilhelms and A. Van Gelder. Octrees for Faster Isosurface Generation. *ACM Transactions on Graphics*, 11(3):201–227, 1992.
10. L.Devroy, L.Gyorfi. *Nonparametric Density Estimation*. Jon Wiley & Sons, 1984.
11. M. Farmen, J.S. Marron. An Assesment of Finite Sample Performace of Adaptive Methods inDensity Estimation. In *Computational Statistics and Data Analysis*, 1998.
12. M.C. Jones M.P. Wand. *Kernel Smoothing*. Chapman & Hall, 1985.
13. W. Lorensen and H. Cline. Marchine cubes: A high resolution 3d surface construction algorithm, 1987.